Knowledge Engineering and Data Science (KEDS)
Vol 2, No 1, Juni 2019, pp. 31–40

pISSN 2597-4602
eISSN 2597-4637

High Dimensional Data Clustering using Self-Organized Map

Ruth Ema Febrita [a, 1], Wayan Firdaus Mahmudy [a, 2, *], Aji Prasetya Wibawa [b, 3]

[a] *Department of Computer Science, Universitas Brawijaya*
Jl. Veteran, Malang, 65145, Indonesia

[b] *Department of Electrical Engineering, State University of Malang*
Jl. Semarang No. 5, Malang 65145, Indonesia

[1] *ruthemaf@gmail.com;* [2] *wayanfm@ub.ac.id*;* [3] *aji.prasetya.ft@um.ac.id*

* *corresponding author*

ARTICLE INFO

Article history:
Received 7 February 2019
Revised 1 April 2019
Accepted 6 April 2019
Published online 23 June 2019

Keywords:
house clustering
k-means
self-organized map
som kohonen

ABSTRACT

As the population grows and e economic development, houses could be one of basic needs of every family. Therefore, housing investment has promising value in the future. This research implements the Self-Organized Map (SOM) algorithm to cluster house data for providing several house groups based on the various features. K-means is used as the baseline of the proposed approach. SOM has higher silhouette coefficient (0.4367) compared to its comparison (0.236). Thus, this method outperforms k-means in terms of visualizing high-dimensional data cluster. It is also better in the cluster formation and regulating the data distribution.

I. Introduction

Houses are promising investment commodity in the last decade. The house price index in Indonesia is also experienced inflation, as well as the level of sales of home property [1]. The price is influenced by several factors such as the interest rate, inflation on house ownership loans, inflation in building materials prices, and inflation in workers' minimum wages. There are many different types of houses offered along with various features, which sometimes make prospective buyers confused to determine their choice.

Another three factors that influence house pricing are physical attributes, accessibility, and developer reputation [2]. Physical attributes refer to house attributes that are visible and measurable, such as the land area, building area, number of rooms, number of bathrooms, and the availability of the living room. The accessibility refers to the house location that determines the ease of access to public facilities, such as hospitals, schools, campuses, markets, etc. Commonly, the closer location of a house with many public facilities may cause the more expensive of the house price. Some other economic phenomena that can affect the house prices are the interest rate, inflation and the Gross Domestic Products (GDP) [3].

Based on many considered features in determining house prices, the housing data are classified as a high-dimensional data. In some previous studies, Neural Network can be used to predict the price of a house [4][5][6][7][8]. Several approaches of regression techniques to predict the house prices also done by [9][10] which using the time-series data. However, in using a neural network or regression techniques, all feature values must be complete, is less applicable in the real condition. It is because the information received from prospective buyers is not always the same and complete.

Although by the neural network the missing input value can be replaced through the interpolation mechanism, when using the interpolation, the replaced value will be given under the assumption that it is related to other variables, which is not always correct in the house pricing case. For example, the first data has a value of 60 meters square of the land area and 30 meters square of the building area, while the second data has a value of 100 meters square of the land area and 50 meters square of the

https://doi.org/10.17977/um017v2i12019p31-40

building area. If the third data has a value of 150 meters square of the land area but has a missing value on building area attributes, then the interpolation will return 75 meters square as the replacement value of the building area attribute, based on the assumption of the two previous data that the building area is half of the land area. The results of interpolation in housing cases are not always correct, because the building area of a house may have a greater value than the land area if the house has more than one floor. Therefore, the use of interpolation in house price predictions can cause inaccurate results.

This study tries to perform a clustering approach to give a recommendation for house prices. The clustering approach may extract the value of features of each cluster, which can be used as a recommendation for house prices. The clustering process is done by comparing all data in the dataset which will then be clustered based on the similarity of existing features. It is expected to provide information on price ranges that are in line with the features which are already known by prospective buyers. Thus, the process of predicting house prices is more applicable. Moreover, if the cluster produced can be easily distinguished from other clusters, the value of the inter-cluster feature will not experience overlapping. This makes it very easy to deal with data which contain a missing value. The price recommendation process can still be done by looking at the value of the known attributes and ignoring the unknown values.

There is some previous research implemented clustering approach to do prediction task in some cases. Two-stage clustering had been for predicting rented house price [11]. The idea of this method is forming the rented house data into some clusters using a clustering algorithm based on the location, then creating the prediction model using linear regression neural network for each cluster formed. The clustering process is done by considering house location because this research believes that the nearer a house location to many public facilities (landmark), the rental price will be higher. By using this hybrid method, the effective cluster can be created, although the accurate rent price prediction still needs more improvements.

The two-stage clustering method using K-Means and Fuzzy Inference System also have been done to cluster house data [12]. The data are clustered into four predefined clusters based on house price: cheap, medium, expensive and very expensive. The clustering method was implemented to see how the location of a house affected the house price. After those clusters have been formed, the values of centroid features from each cluster were obtained and used as initial values to build a fuzzy inference system. This research shows that the fuzzy clustering system cannot predict the same cluster as K-means, means that the prediction of the house price still low in the accuracy. Similar work also had been done by [13], which tried to predict house prices using three different methods such as Fuzzy, Artificial Neural Network and the K-NN.

Another clustering method, Fuzzy C-Means, also been used as a hybrid method in predicting cases. A hybrid method of Fuzzy C-Means and regression technique is used to predict the workload of a new driver [14]. Fuzzy C-Means was used to generate a driver workload model based on the regression generated previously. Meanwhile, [15][16] also developed Fuzzy C-Means for predicting the software fault by using it as feature extraction method.

On the other hand, SOM had been applied to classify and label transient data signal [17]. The sequence of stable and transient phase is extracted from the time-series signal data obtained from aircraft engines during the flights. SOM cluster and label the transient data by checking the similarity of the pattern. The accuracy of the labeled transient signal is excellent in robustness and visualization.

A Generalized SOM (GSOM) is the improvement of SOM, have been studied [18]. The special characteristic of this method is it can automatically determine the best number of the cluster and also the shape of the cluster by using a 1-D neighborhood method. The 1-D neighborhood method was represented like the chain of neurons, which can automatically disconnect and connect with the other neurons.

SOM has also been implemented in the health sector to classify and predict female subject with unhealthy visceral fat levels in Japan [19]. A map topology is formed from the neurons, where each neuron stores 13 health parameters that are used to detect visceral fat. This map topology is then trained using the SOM algorithm, and each neuron will be given a label that represents the visceral fat level. The test data prediction is done by finding a winning neuron, which is a neuron that stores the value of the closest / most similar feature to the data.

Self-Organized Map (SOM) Kohonen will be implemented to cluster high-dimensional data of housing data. SOM was proposed since it works based on topological arranged neurons, where each neuron has a different feature value. SOM also has a neighboring weight updating mechanism, which causes adjacent neurons to have similar characteristics. In other words, it is expected to improve the cluster visualization. This research uses K-means as a comparison approach. The performance of these methods are compared to discover the best algorithm for data high dimensional house-data clustering.

II. Methods

A. Dataset

The dataset consists of 189 housing data which are obtained from the property exhibition, held in March and August 2017. All data in the dataset have a different value of physical attributes, locations, and also have valid prices, are determined by the developer and valid until December 2017. This research uses [20] to obtain the exact number of public facilities around the house location, within a 1000 meter radius. All the feature values will be normalized to optimize the clustering process, All features that build this house dataset will be shown in Table 1. Meanwhile, the complete dataset can be accessed through [21].

B. SOM Clustering

SOM is one type of neural network, which is categorized as an unsupervised algorithm. SOM is built by using one or more layer of neurons and can be described as a topological map of neurons. In general, the SOM algorithm works by finding a neuron that has the most similar weight corresponding to the data, which is then called as the winning neuron, and then updates the weights of the surrounding neurons within the neighboring radius to form the cluster of neurons that have similar weights. The applied SOM algorithm is detailed as in [22]:

- *Initialization*. In this first step, some of SOM parameters, such as vector weight of neurons, the map size, the learning rate and also the radius of neighborhood update (N_c) need to be initialized. The two-dimension rectangular map grid will be used in this research, while the size of the map will be tested to obtain the best size which performs the best clustering result. Meanwhile, each neuron contains a set of features value which already described in Table 1. The learning rate represents how fast the algorithm will learn in each iteration. The radius of the neighborhood update refers to the number of neurons around the winning neuron that will be updated.
- *Obtaining the winner neuron*. Each data vector (x) in the dataset will be compared to each neuron weights (w_i) contained in the topological map, and the data similarity (d) will be calculated by using the Euclidean distance, as written in (1). The neuron that has the closest distance to the data will be called the winning neuron (c).

$$d = \sqrt{\sum_{i=1}^{n}\left(x_i - w_{ij}\right)^2} \tag{1}$$

- *Neighborhood weights update*. This step is an effort to make the weight of the adjacent neurons have similar weights. Updating the weights is done using the equation (2) and (3).

$$w_{ij}(t+1) = w_{ij}(t) + h_{ci}(t)\left[x(t) - w_{ij}(t)\right] \tag{2}$$

$$h_{ci} = h_0 exp(-\|r_i - r_c\|^2/\sigma^2) \tag{3}$$

$h_{ci}(t)$ is the learning rate $\alpha(t)$ for all neurons within the N_c and $h_{ci}(t)$= 0 for all neurons outside the N_c. r_i and r_c is the weight of neuron i and the winner neuron c, $\sigma = N_c$. The distance of neuron i and neuron c ($\|r_i - r_c\|$) is calculated based on the neurons positions in the grid map.

- *Stopping criteria*. The stopping criteria are determined by using (4), where e is the minimum allowable weight change of the neuron weights between the coressponding iteration (t) and the previous iteration (t-1)

$$e = \sum \frac{\Sigma\left(w_{ij}^{t-1} - w_{ij}^{t}\right)}{n \times m} \tag{4}$$

Table 1. The list of observed attributes of a house data

Attributes	Original Units
Regency ID	-
House ID	-
Distance from KM 0 b	Kilometer
Building area	Meter square
Land area	Meter square
Number of hospital	Item
Number of clinic / pharmacy	Item
Number of schools	Item
Number of campuses	Item
Number of market / mall	Item
Number of hotels	Item
Number of restaurants	Item
Number of recreational park	Item
Number of public transportation	Item
Number of worship place	Item
Number of bedrooms	Item
Number of bathrooms	Item
Living room	Meter square
Family room	Meter square
Kitchen	Meter square
Dinning room	Meter square
Clothes horse	Meter square
Number of floor	Item
Warehouse	Meter square
Garage	Meter square
Number of terace/ balcony	Item
Garden	Meter square
Swimming pool	Meter square
Building permission	-
Electrical installation	-
Water channel	-
Certificate of ownership	-
Free fence	-
Free kitchen set	-
The cost of making land certificate	-
housing ownership credit interest rates	%
Price	IDR

*km 0 in the Malang City is in Malang Square (Merdeka Selatan Street)

C. Defining the Cluster

There are several assumptions used in defining clusters. For convenience, the topological map will be described in Figure 1 where each cell describes a neuron and the number written in each cell illustrate the amount of data that best matches the weight of the neuron.. The more detailed illustration will be explained as follows:

- The cluster should have at least two matching data in one of the neurons or it may have only one matching data with other data in the adjacent neurons. In Figure 1a, the red color has the insufficient condition to make a cluster.
- If there are two cells or additional cells separated by empty cells (not adjacent), then every cell is going to be thought-about as a special cluster (Figure 1b).
- If there are two or more adjacent cells, which have the matching data on them, then all of the adjacent cells will be considered as the same cluster (Figure 1c).

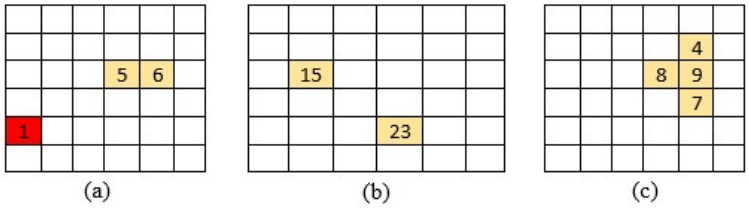

Fig. 1. The illustration of defined cluster; (a) the minimum condition of a cluster; (b) a special cluster; (c) a cluster with two adjacent cells

Thus, the cells that do not have any numbers describe the neurons that have no compatibility with any data in the dataset.

D. The Measurement of Cluster Validity

In order to measure how well the results of a clustering process, the Silhouette coefficient and Davies-Bouldin Index are used. The principle of measuring silhouette coefficient is that a cluster is good enough if the distance between members in the same cluster is close, while the distance between two clusters is far enough so that each cluster can be easily recognized and separated from the other clusters. The Davies-Bouldin Index is used to evaluate cluster results by measuring the ratio of the spread of clusters and the distance between clusters. The Silhouette coefficient will be shown at (5), while the Davies-Bouldin Index will be shown at (6) to (8).

$$\text{Sil} = \text{b} - \text{a} / \max(\text{a}, \text{b}) \tag{5}$$

In (5), a is the mean intra-cluster distance, whereas b is the nearest-cluster distance. The value of Silhouette coefficient will be in the range of [-1, 1]. The most effective cluster will be obtained if the value of Sil=1, therefore the worst value of the Silhouette coefficient is Sil=-1. When the value of Sil=0, it indicates that the clusters are overlapped.

In the Davies-Bouldin Index, the distribution of clusters will be calculated using (6).

$$S_i = \frac{1}{T_i}\sum_{x \in C_i} \|x - z_i\| \tag{6}$$

T_i is the number of members in the cluster i (C_i) and Z_i is the center of the cluster i. The distance between clusters is calculated using the Euclidean distance between the centroid of the cluster i and the centroid of cluster j. The ratio between C_i and C_j will be calculated using (7).

$$R_{ij} = \left\{\frac{S_i + S_j}{d_{ij}}\right\} \tag{7}$$

Then, the maximum value of the ratio (D) will be used to calculate the Davies-Bouldin Index, which is shown at (8).

$$DBI = \frac{1}{K}\sum_{i=1}^{K} D_i \tag{8}$$

Unlike the Silhouette coefficient, Davies-Bouldin Index (DBI) value has a range between [0 - 1]. $DBI = 0$ indicates that the ratio of data distribution in clusters is very good, while $DBI = 1$ shows the ratio of data distribution in clusters is very bad.

III. Results and Discussions

A. SOM Parameter Testing

There are some parameters on SOM that will be tested in this research. For each value in the parameter testing will be tested by 7 times. The first tested parameter is the radius of the neighborhood update (N_c). This parameter serves to determine the area of weight updates in the neurons located around the winning neurons. The closer the neuron position to the winning neuron, the more significant weight changes will be so that the weight of the neuron will be more similar to the weight of the winning neuron. When testing the neighboring radius, other parameters are temporarily set by default with the following values: the map size of the neuron is 15×15, the learning rate is α = 0.05,

and the maximum error is = 0.1. The percentage of values will be used for this parameter testing. For example, if the map size is 15×15 and the N_c=60%, that means the radius of the neighborhood update is N_c=9 (9 neurons above, 9 neurons on the left side, 9 neurons below and 9 neurons on the right side of the winning neurons). Table 2 will show the result of this parameter testing.

Table 2 shows that for each neighboring radius tested, the value of Silhouette coefficient always shows a negative value. This negative value of silhouette coefficient can be influenced by other parameters. The best Silhouette coefficient value is obtained when the neighbor radius is 67% of the map size. The best DBI value (the smallest DBI value) is also obtained when the neighbor's radius size is 67%. The test results show that when N_c=67%, the resulting cluster has a good ratio in terms of the number and distance between clusters, but still does not perform an effective cluster. The size of the neighboring radius that is too large (80%) causes the cluster boundaries to be less clear because the area of the neuron whose weight will be updated is too broad. This will allow the weight of a neuron look similar to that of one cluster. In the other hand, when the size of the neighboring radius is too small will cause the cluster formation process will last very slow. In testing other parameters, the neighboring radius will be set to 67% of the map size.

The next tested parameter is the learning rate (α). The result of the learning rate will be shown in Table 3. Based on the tests performed, the best silhouette coefficient is 0.0958, which is obtained at the setting α = 0.06. While the best DBI is obtained at α = 0.05. All of the tested learning rate values show that the algorithm just needs 3-4 iterations for running the clustering process. This fact shows that the learning rate does not affect the number of iterations. By considering the average Silhouette coefficient, the average DBI, the best Silhouette coefficient, and the best DBI value, the next parameter testing will use the learning rate α=0.06.

The following tested parameter the maximum error (e) as the stopping criteria in the clustering process. The maximum error in SOM testing can be considered as a significant change in weight on neurons compared to the weights in the previous iteration. The test result of the stopping criteria will be shown in Table 4.

Table 2. Neighborhood radius testing

Nc (%)	Average of Silhouette coef	Best of Silhouette coef	Average of DBI	Best of DBI	Number of Cluster
13	-0.884	-0.566	0.7218	0.1461	2
27	-0.927217	-0.8611686	0.587514	0.1594	2
40	-0.951212	-0.7657426	0.827013	0.2678	3
53	-0.959121	-0.8336872	0.795857	0.18	4
67	-0.739308	-0.4496412	0.552214	0.106	2
80	-1	-1	1	1	1

Table 3. Learning rate testing

α	Average of Silhouette coef	Best of Silhouette coef	Average of DBI	Best of DBI	Number of Cluster
0.01	-0.88693	-0.41373	0.764971	0.3149	2
0.02	-1	-1	1	1	1
0.03	-0.91634	-0.41435	0.871257	0.0988	1
0.04	-0.82412	-0.39095	0.758229	0.1148	2
0.05	-0.85469	0.017196	0.868429	0.079	2
0.06	-0.77649	0.0958	0.791271	0.0979	2
0.07	-0.7262	-0.27805	0.615897	0.08138	2
0.08	-0.94572	-0.77001	0.885571	0.268	2
0.09	-0.86777	-0.61331	0.52072	0.04434	2
0.1	-0.94549	-0.61842	0.769143	0.12	4
0.2	-0.9291	-0.50367	0.878	0.146	3
0.3	-0.99687	-0.97811	0.766324	0.13327	3
0.4	-0.948	-0.72479	0.8899	0.4195	3

In the stopping criteria, the smaller the error value specified, the more similar the weights of the map from the current iteration with the previous iteration. Based on the test results, the greater the error value results in the fewest iterations needed for a clustering process. The test results also show that at e=0.45 and e=0.5 there is increasing value of Silhouette coefficient and DBI, both on average and the best value. This can happen because the clustering process will immediately stop the program when a very significant change in weight occurs, whereas in this condition, there has not been a lot of data transfer from one neuron to another, so the data is still quite scattered. When the clusters are quite diffuse, it is possible for the clustering result to obtain the better Silhouette coefficient value as well as the DBI value. Based on the testing, the best stop criteria occur when e=0.5, because it shows the best average value of Silhouette coefficient and DBI. However, the best silhouette coefficient values are obtained when e=0.4. Thus the stop condition is set with the value e = 0.5 to for the next parameter testing.

The last testing parameter is the size of the topological map. The test result is shown in Table 5. The test results show that the best map size is 30×30. When map size is 10×10 shows the worst results because the number of neurons in it is much smaller than the training data used, which is 189 house data. Thus, a neuron can be matched with a lot of home data so that a good cluster is very difficult to achieve. The 10×10 maps will provide very limited distances to make separate distances between clusters. As consequence, the Silhouette coefficient will be very small.

After doing several parameter testing, the best number of clusters obtained by using SOM is n=2, although in some testing the number of clusters can reach up to n=4. The visualization of the best clustering result is shown in Figure. 2

B. K-Means Result

The following sub-section will discuss the implementation of other clustering algorithms as a comparison of the SOM algorithm. Unlike the SOM algorithm, the number of clusters in the K-Means algorithm must be specified before testing. The parameters that will be tested in K-Means are the number of clusters and also the stopping criteria (e). Table 6 and Table 7 will show the testing result of the K-Means algorithm based on the Silhouette coefficient and DBI. All values written in the table are the best values for each group testing.

Table 4. Maximum error testing

e	Average of Silhouette coef	Best of Silhouette coef	Average of DBI	Best of DBI	Number of Cluster
0.01	-0.9889	-0.92231	0.883414	0.1839	3
0.05	-0.79485	-0.4469	0.532481	0.237	2
0.1	-0.55797	0.03704	0.558784	0.0712	2
0.15	-0.61825	0.052803	0.568891	0.07284	2
0.25	-0.67872	-0.11946	0.548464	0.10143	2
0.3	-0.77888	0.382642	0.756946	0.04852	2
0.35	-0.69045	0.087995	0.688443	0.0835	2
0.4	-0.80267	0.381318	0.8683	0.0781	2
0.45	-0.28458	0.436725	0.401703	0.0513	2
0.5	-0.31293	0.250545	0.463836	0.0491	2

Table 5. Map size testing

Map Size	Average of Silhouette coef	Best of Silhouette coef	Average of DBI	Best of DBI	Number of Cluster
10 x 10	-0.9786	-0.8502	0.871986	0.1039	2
15 x 15	-0.86143	-0.37508	0.619	0.197	2
20 x 20	-0.55797	0.03704	0.558784	0.0712	2
25 x 25	-0.84044	-0.34323	0.653924	0.12244	2
30 x 30	-0.48226	0.351387	0.151423	0.05173	2
35 x 35	-0.67708	0.019051	0.583639	0.2163	2

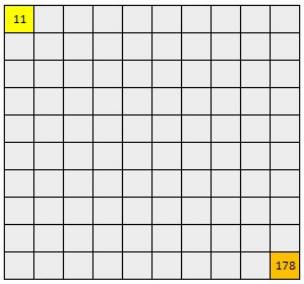

Fig. 2. The best clustering visualization using SOM

Based on tests performed using K-Means, the best Silhouette is obtained when the number of clusters specified as n = 6 and e = 0.5. However, the number of clusters that actually formed as the clustering result does not exceed n = 3. In testing the K-Means method, almost all of the parameter values tested performs a negative Silhouette coefficient value. This indicates that the resulting cluster is not right and still difficult to distinguish between one cluster and another. The best number of clusters obtained is 3 clusters.

C. SOM and K-Means Comparison

Based on the test results, both SOM and K-Means are still difficult to achieve good clustering results. It is proven by the negative value of the average Silhouette coefficient, indicates that many data are send to the wrong cluster. However, the best Silhouette coefficient achieved by SOM (0.4367), is better than K-Means (0.236). In this, case the SOM has a better ability to build valid clusters compared to the K-Means. SOM algorithm represents the data in the form of a two-dimensional topology map. In SOM, data is placed on neurons. As a result, the internal distance among the member of the cluster and the distance between clusters is easier to be measured.

In terms of data distribution, SOM shows better performance compared to K-Means. In SOM, each cluster can be clearly identified by searching for the grid distance between clusters on the map, so that the distance between clusters can be calculated easily and clearly. In clustering the data, SOM compares input data vectors to the weights of neurons. But in the K-Means, input data vector is compared to the value of the centroid. The centroid values in K-Means always been updated on each iteration with the average value of its members' features. Thus, a centroid does not always indicate a point in the dataset. This may create difficulties in determining the cluster area and cluster distribution.

Although SOM shows better performance compared to K-Means, for clustering the high-dimensional data it still needs more improvements. This is because in the SOM, in determining the winning neuron is done by calculating the similarity between the data and the weight of the neurons

Table 6. K-means clustering result based on the silhouette coefficient

N-Cluster	Maximum Error (*e*)				
	0.01	**0.05**	**0.1**	**0.2**	**0.5**
C=3	-0.114403	-0.438083	-0.398426	-0.374	0.072083
C=4	-0.481474	-0.357111	-0.195223	-0.05731	-0.21398
C=5	-0.459957	-0.180032	-0.614222	-0.00155	-0.45512
C=6	-0.268694	-0.855462	0.1104888	0.06809	0.236027

Table 7. K-means clustering result based on the DBI

N-Cluster	Maximum Error (*e*)				
	0.01	**0.05**	**0.1**	**0.2**	**0.5**
C=3	0.4973335	0.4515325	0.4848922	0.408955	0.51021
C=4	0.3420642	0.4083031	0.4619643	0.423122	0.529518
C=5	0.4847796	0.5156256	0.449523	0.608689	0.553155
C=6	0.4619643	0.2885393	0.2954057	0.513923	0.516404

by using the calculation of the Euclidean distance. In calculating the Euclidean distance, all features are calculated using the same weight. Whereas, in high-dimensional data, not all features are relevant. This makes considering all features in the calculation can actually be a disruption to form a valid clustering result. In fact, some features in high-dimensional data can be referred as the noise [23]. Considering the characteristic of the data, SOM can be modified by using different distance measurements, for example using Manhattan distance [24][25][26].

IV. Conclusion

SOM can be used to cluster housing data and successfully shows better performance compared to the K-Means algorithm. SOM outperforms K-Means in terms of visualizing the high-dimensional data clustering. In other words, it provide easier calculation to obtain the cluster validity. In addition, SOM also showed a better performance in the process of forming good clusters, is indicated by obtaining better Silhouette coefficient and DBI values. However, SOM still needs some improvements to produce better clustering results.

Acknowledgement

The authors would like to thank Andrini Cahyaningrum Pratiwi for helping in the data collection stage. In addition, the authors also wishes to acknowledge to Oliver Beattie for providing the open accessed interface which was used in this research.

References

[1] Bank of Indonesia, "Residential Property Price Survey," 2017. [Online]. Available: http://www.bi.go.id/id/publikasi/survei/harga-properti-primer/Pages/SHPR-Tw.IV-2016.aspx. [Accessed: 14-Mar-2017].

[2] R. A. Rahadi, S. K. Wiryono, D. P. Koesrindartotoor, and I. B. Syamwil, "Factors influencing the price of housing in Indonesia," *International Journal of Housing Market Analysis*, vol. 8, no. 2, pp. 169–188, 2015.

[3] R. Füss and J. Zietz, "The economic drivers of differences in house price inflation rates across MSAs," *Journal of Housing Economics*, vol. 31, pp. 35–53, 2016.

[4] W. T. Lim, L. Wang, and Y. Wang, "Singapore Housing Price Prediction Using Neural Networks," *12th International Conference on Natural Computation, Fuzzy System and Knowledge Discovery*, pp. 518–522, 2016.

[5] Y. Feng and K. Jones, "Comparing multilevel modelling and artificial neural networks in house price prediction," *2015 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services*, pp. 108–114, 2015.

[6] J. J. Wang *et al.*, "Predicting House Price With a Memristor-Based Artificial Neural Network," *IEEE Access*, 2018.

[7] Y. Yu, S. Song, T. Zhou, H. Yachi, and S. Gao, "Forecasting House Price Index of China Using Dendritic Neuron Model," *2016 International Conference on Progress in Informatics and Computing*, pp. 37–41, 2016.

[8] A. Varma *et al.*, "House Price Prediction Using Machine Learning And Neural Networks," *2018 Second International Conference on Inventive Communication and Computational Technologies*, pp. 1936–1939, 1936.

[9] S. Lu, Z. Li, Z. Qin, X. Yang, R. Siow, and M. Goh, "A Hybrid Regression Technique for House Prices Prediction," *IEEE International Conference on Industrial Engineering and Engineering Management,* pp. 319–323, 2017.

[10] F. Tan, C. Cheng, and Z. Wei, "Time-aware Latent Hierarchical Model for Predicting House Prices," *International Conference on Data Mining*, pp. 1111–1116, 2017.

[11] Y. Li, Q. Pan, T. Yang, and L. Guo, "Reasonable price recommendation on Airbnb using Multi-Scale clustering," *Chinese Control Conference CCC*, vol. 2016–Augus, pp. 7038–7041, 2016.

[12] R. E. Febrita, A. N. Alfiyatin, H. Taufiq, and W. F. Mahmudy, "Data-driven Fuzzy Rule Extraction for Housing Price Prediction in Malang , East Java," *9th International Conference on Advance Computer Science and Inference Systems*, 2017.

[13] M. F. Mukhlishin, R. Saputra, and A. Wibowo, "Predicting House Sale Price Using Fuzzy Logic, Artificial Neural Network and K-Nearest Neighbor," *1st International Conference on Informatics and Computational Sciences (ICICoS),* vol. 1, pp. 171–176, 2017.

[14] D. Yi, J. Su, C. Liu, and W. Chen, "New Driver Workload Prediction Using Clustering-Aided Approaches," *IEEE Transaction on Systems, Man, and Cybernetics Systems*, vol. 49, no. 1, pp. 64–70, 2019.

[15] A. Arshad, S. Riaz, L. Jiao, and A. Murthy, "Semi-Supervised Deep Fuzzy C-Mean Clustering for Software Fault Prediction," *IEEE Access*, vol. 6, pp. 25675–25685, 2018.

[16] A. Arshad, S. Riaz, L. Jiao, and A. Murthy, "The Empirical Study of Semi-Supervised Deep Fuzzy C-Mean Clustering for Software Fault Prediction," *IEEE Access*, vol. 6, pp. 47047–47061, 2018.

[17] C. Faure, M. Olteanu, J. M. Bardet, and J. Lacaille, "Using self-organizing maps for clustering and labelling aircraft engine data phases," *12th International Workshop on Self-Organizing Maps Learning Vector Quantization, Clustering Data Visualization WSOM 2017 - Proceeding*, 2017.

[18] M. B. Gorzalczany and F. Rudzinski, "Generalized Self-Organizing Maps for Automatic Determination of the Number of Clusters and Their Multiprototypes in Cluster Analysis," *IEEE Transaction on Neural Networks and Learning Systems*, pp. 1–13, 2017.

[19] N. Kamiura, S. Kobashi, M. Nii, T. Yumoto, and K. Sorachi, "Application of self-organizing maps to data classification and data prediction for female subjects with unhealthy-level visceral fat," *2016 IEEE International Conference on Systems, Man, and Cybernectics*, pp. 001815–001820, 2016.

[20] "Draw Radius Circles on A Map." [Online]. Available: http://obeattie.github.io/gmaps-radius/?

[21] R. E. Febrita, "Published House Dataset," 2018. [Online]. Available: wayanfm.lecture.ub.ac.id/files/2018/10/Published-Dataset-Ruth-Ema.xlsx.

[22] T. Kohonen, "The self-organizing map," *Proceeding IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[23] C. Peng, Z. Kang, M. Yang, and Q. Cheng, "Feature Selection Embedded Subspace Clustering," *IEEE Signal Processing Letters*, vol. 23, no. 7, pp. 1018–1022, 2016.

[24] A. B. Rathod, "A Comparative Study on Distance Measuring Approches for Permutation Representations," *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*, pp. 251–255, 2016.

[25] L. Greche, M. Jazouli, N. Es-sbai, A. Majda, and A. Zarghili, "Comparison Between Euclidean and Manhattan Distance Measure for Facial Expressions Classification," *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, pp. 1–4, 2017.

[26] J. P. Singh and N. Bouguila, "Proportional Data Clustering using K-Means Algorithm : A comparison of different distances," *2017 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1048–1052.